

《C/C++ 学习 指南》

第26.1讲：类的朋友

作者：邵发 QQ群：417024631

官网：<http://www.afanihao.cn/>

习题：<http://www.afanihao.cn/kbase/>

本书配套纸质教材，具体信息见官网。

C/C++学习指南 邵发 www.afanihao.cn

引 例

```
class Object
{
public:
    Object(int v) : value(v)
    {
    }
private:
    int value;
};
```

```
void Print(Object* p)
{
    printf("value: %d \n", p->value);
}
```

显然，在全局函数Print中无法访问Object对象的私有成员

类的朋友

在类A中，用friend关键字可以将一个全局函数声明为该类的“朋友”。也可以将另一个类声明为该类的“朋友”。

```
class Object
{
    // 将全局函数Print声明为“朋友”
    friend void Print(Object* p);

    // 将类Something声明为“朋友”
    friend class Something;
};
```

原则上说，该函数、类只是这个类的“朋友”，而非“成员”。（有些教程上称之为友员、朋友成员）

类的朋友

(1) friend声明语句：位置自由，一般放在类体大括号的最前头。

（注意区分类的声明和类的定义，这里只是声明一下）

(2) 朋友拥有的权利：

可以无限地访问该类的所有成员

不受protected/private的限制

（注意：朋友关系是单向的，Something类可以自由访问Object类，但反过来不成立）

C/C++学习指南 邵发 www.afanihao.cn

举例

```
class Object
{
    friend void Print(Object* p);
    ....
};
void Print(Object* p)
{
    printf("value: %d \n", p->value);
}
```

C/C++学习指南 邵发 www.afanihao.cn

举例

```
class Object
{
    friend class Something;
    ....
};
class Something
{
public:
    void Print(Object* p)
    {
        printf("value: %d \n", p->value);
    }
}
```

应用场景

负面作用： friend是不常用的语句，它破坏了访问修饰符的作用，对类的封装够成威胁。

正面作用： 可能的应用场景：在封装一个模块时，模块中有多个类，那么他们之间可以互相声明为朋友以便操作（他们是一伙的）。

也就是说，模块内部的类都是朋友，而外部调用者不是朋友、受访问修饰符的限制。

小结

1. 学习朋友的声明语法
2. 该语法不常用，介绍了可能的应用场景